International Macro and Trade Assignment 2

Tanya Rajan

October 26, 2020

Table 1: Estimate a regression of log (non-zero) trade flows on log bilateral distance, a contiguity indicator, a common-language indicator, exporter-year fixed effects, and importer-year fixed effects for the years 2000-2006. When using reg, you will have to create the fixed-effect dummies. When using the xtreg, areg, and reghdfe commands, use the fixed-effect options to absorb them. Report the computation time associated with each estimator.

Table 1 below shows the estimates and computational times (in seconds) of each of the four fixed effects methods in Stata.

Table 1: Fixed Effect Estimations Across Specifications

	(1) **reg**	(2) **xtreg**	(3) **areg**	(4) **reghdfe**
Log Distance	-1.658	-1.658	-1.658	-1.658
	(0.00875)	(0.00875)	(0.00875)	(0.00875)
Time	135.669	36.015	29.288	1.305
N	156248	156248	156248	156178
R^2	0.718	0.586	0.718	0.718
Adj. R^2	0.713	0.578	0.713	0.713

Standard errors in parentheses

1. Are the point estimates and standard errors numerically identical across the different estimators? Should they be?

Yes, the point estimates and the standard errors are identical, as can be seen in Table 1 above. They should be identical since all of the specifications being used are equivalent. The only difference between each column is the computational efficiency of the package being used, which is reflected in the time reports at the bottom of the table.

2. Are the number of observations and R-squared statistics identical? Should they be?

The number of observations are the same in all the packages except **reghdfe**. In the documentation for the **reghdfe** package, the authors talk about the danger of including singleton groups with only one observation per group.[1] As a result, they drop singleton groups in the **reghdfe** estimation. I verify this by manually creating indicators for singleton observations within the exporter-year and importer-year groups, and seeing

[1] See http://scorreia.com/research/singletons.pdf for a discussion

that the sample drops down to 156,178 when I exclude these observations.

The R^2 estimates are the same for all models other than **xtreg**. This is the case because **xtreg** assumes that group effects are fixed, so it essentially leaves out the contribution of the fixed effects to the fit of the model when calculating R^2, while the other methods include them. It makes sense, then that **xtreg** repots a lower R^2 indicating a "poorer" model fit.

> 3. How do the relative computation times of these estimators depend on the dimensionality and size of the data?

When running the same functions on a smaller subset of the data (only including the year 2006), the order of times from slowest to fastest is **xtreg** > **reg** > **areg** > **reghdfe**. When running on a larger dataset (e.g. including all years), **reghdfe** performs the best while **reg** is sometimes unable to compute the fixed effect specification due to memory issues.

> Table 2: Now we examine the roles of zeros in the trade-flow matrix. Run a regression for the years 2004-2006 with the same covariates as above.

Table 2 below lists the estimation results and times for various estimation specifications. The specification details are described below the results.

Note: In specification (4), fixed effect indicators had to be created as new variables in order to input them (one couldn't simply use the "i.variable" command in Stata). As a result, estimation took a long time. Additionally, the variance matrix was highly singular so the model does not have calculable standard errors.

	(1) **reghdfe**	(2) **reghdfe**	(3) **reghdfe**	(4) **ppml**	(5) **poi2hdfe**	(6) **ppml_panel_sg**	(7) **ppmlhdfe**	(8) **ppmlhdfe**
Log Distance	-1.704	-1.069	-0.890	-0.899	-0.899	-0.899	-0.899	-0.899
	(0.0136)	(0.00756)	(0.00652)	(.)	(0.0157)	(0.0157)	(0.0157)	(0.0157)
Contiguity	0.971	1.017	1.118	0.462	0.462	0.462	0.462	0.464
	(0.0617)	(0.0343)	(0.0320)	(.)	(0.0377)	(0.0377)	(0.0377)	(0.0376)
Common Language	0.975	0.470	0.433	0.212	0.212	0.212	0.212	0.211
	(0.0286)	(0.0159)	(0.0132)	(.)	(0.0358)	(0.0358)	(0.0358)	(0.0357)
N	67,350	67,350	91,652	91,685	91,652	91,685	91,652	67,350
R^2	0.718	0.758	0.742	0.907		0.907		
Time	.739	.402	.528	448.501	15.007	13.432	4.547	2.541
Include Zeroes?	No	No	Yes	Yes	Yes	Yes	Yes	No
Method	Log-Linear	Log-Linear	Log-Linear	Poisson	Poisson	Poisson	Poisson	Poisson
log(Flow + 1)?	No	Yes	Yes	No	No	No	No	No

Standard errors in parentheses

> 1. Are your results sensitive to the omission of zeros?

Somewhat. When we go from specification (2), which is log-linear with $\ln(\text{flow}+1)$ and has no zeroes, to specification (3), which is log-linear with $\ln(\text{flow}+1)$ including zeroes, we see that the effect of log distance on log flow decreases in magnitude. However between specifications (7) and (8), which are both Poisson specifications, we see that excluding zeroes does not change the estimates much.

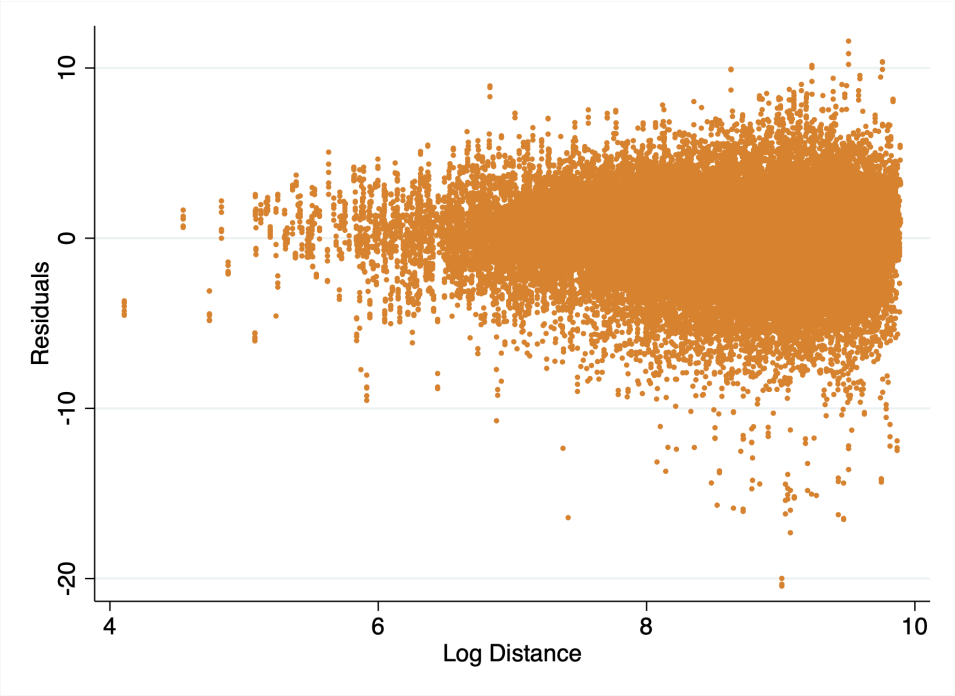
> 2. How well does making the dependent variable log(x+1) perform?

Changing from specification (1) to (2) by using log(x+1) as the dependent variable results in a change of estimates and standard errors. It performs well when we use it in order to include zeroes in the log-linear specification. The coefficient on log distance in specification (3), which is log-linear and uses log(flow + 1), aligns well with the estimates from the Poisson specifications.

> 3. Examine the residuals from your log-linear regression. Are they heteroskedastic? Report a Breusch–Pagan test statistic and a scatterplot of the residuals that addresses this question.

The residuals do seem heteroskedastic. The Breusch-Pagan test statistic from the log-linear specification is 8415.97. Additionally, we can visually inspect the figure below to see that greater log distance values generally have residuals with higher variance than smaller log distance values. This suggests heteroskedasticity.

Figure 1: Residuals by Log Distance



> 4. How do the computation times compare?

Since we established in Table 1 that **reghdfe** was the fastest of the log-linear estimators and gives the same results as the other packages, I choose to use only **reghdfe** for the log-linear specifications in Table 2. As a result, the log-linear specifications are the fastest, which makes sense since the Poisson methods use maximum likelihood and require optimization to find the estimates. The **ppml** specification is the slowest while **ppmlhdfe** is the fastest of the Poisson methods.

> Table 3: Now estimate the log-linear specification of Table 1 using non-zero trade flows for all years (1948-2006) using reghdfe. Compare the speed of this calculation to the speed of estimating it in Julia using the FixedEffectModels package and in R using the fixest package. Use heteroskedastic-robust standard errors in all cases.

I output the results of the estimations using **esttab** in Stata, **etable** in R, and **regtable** in Julia, which can be found in my submission zip file. To better compare results, I reconcile table formats across these three

packages and combine the results into a single table below. Given the results of the last question, I now use the robust standard errors in all of the specifications reported below.

Statistical Software:	R	Stata	Julia
Log Distance	-1.325	-1.325	-1.325
	(0.0038)	(0.0038)	(0.0038)
Contiguous	0.550	0.550	0.550
	(0.0160)	(0.0160)	(0.0160)
Common Language	0.762	0.762	0.762
	(0.0077)	(0.00774)	(0.00774)
Observations	709,573	709,248	709,248
Time	0.213	5.143	0.7695

White-corrected standard-errors in parentheses

1. Verify that **reghdfe**, **FixedEffectModels**, and **fixest** return identical estimates. Are the standard errors identical?

As can be seen in the table above, all the packages return the same estimates. The standard errors, which are all heteroskedasticity-robust, are identical across specifications as well. We would expect this to be the case since we specified the same options in each of the different packages and languages.

2. Which estimator is faster? By what magnitude?

The R package **fixest** is the fastest. Julia is about 3.3 times slower than R while Stata is about 24 times slower than R and 6.5 times slower than Julia. Note: I ran the **FixedEffectModels** specification on a smaller sample (years 2004-2006) in order to let it compile the function. I timed the package's performance on its second run when I fed it the whole dataset.